CLAIMS

1. A network interface device, comprising:

host interface logic, arranged to receive from a host processor a frame of outgoing data that includes outgoing header information and outgoing payload data, and to separate the header information from the payload data;

an outgoing data memory, coupled to receive the outgoing payload data from the host interface logic;

an outgoing header memory, coupled to receive the outgoing header information from the host interface logic;

a transmit protocol processor, coupled to read and process the outgoing header information from the outgoing header memory so as to generate at least one outgoing packet header in accordance with a predetermined network protocol; and

transmit logic, coupled to receive and associate the at least one outgoing packet header with the outgoing payload data from the outgoing data memory, so as to generate at least one outgoing data packet for transmission over a network in accordance with the protocol.

2. A device according to claim 1, wherein the protocol comprises a network layer protocol.

3. A device according to claim 2, wherein the network layer protocol comprises an Internet Protocol (IP).

4. A device according to claim 1, wherein the protocol comprises a transport layer protocol.

5. A device according to claim 4, wherein the transport layer protocol comprises a Transport Control Protocol (TCP).

6. A device according to claim 4, wherein the transport layer protocol comprises a User Datagram Protocol (UDP).

7. A device according to claim 1, wherein the outgoing data memory and the outgoing header memory comprise parallel first-in-first-out (FIFO) buffers, which are arranged to hold the outgoing payload data and outgoing header information, respectively, for a plurality of frames of outgoing data.

8. A device according to claim 7, wherein the outgoing header memory comprises a first FIFO buffer, coupled to hold the header information before it is processed by the protocol processor, and a second FIFO buffer, coupled to receive the at least one packet header from the protocol processor and to deliver it to the transmit logic.

9. A device according to claim 1, wherein the outgoing header memory comprises a fast memory, coupled to the transmit protocol processor so as to be accessed thereby in a single clock cycle of the processor.

10. A device according to claim 1, wherein at least the outgoing data and header memories and the transmit logic are contained together with the transmit protocol processor in a single integrated circuit chip, and wherein the transmit protocol processor is coupled to the host interface logic so as to enable reprogramming of the transmit protocol processor.

11. A device according to claim 1, wherein the at least one outgoing packet header comprises a plurality of

outgoing packet headers, and wherein the transmit logic is arranged to associate each of the outgoing packet headers with a corresponding portion of the outgoing payload data so as to generate a sequence of outgoing data packets.

12. A device according to claim 1, wherein the at least one outgoing packet header comprises a data length parameter, and wherein the transmit logic is arranged to read the data for inclusion in the outgoing packet responsive to the data length parameter.

13. A device according to claim 1, and comprising:

receive logic, which is coupled to receive from the network an incoming data packet comprising incoming data that includes an incoming header and incoming payload data, and to select a header portion of the incoming data packet;

an incoming header memory, coupled to receive from the receive logic a header portion of the incoming data, which includes at least the incoming header;

an incoming data memory, coupled to receive from the receive logic a data portion of the incoming data, which includes at least the incoming payload data; and

a receive protocol processor, coupled to read and process the header portion of the incoming data in accordance with the predetermined network protocol so as to generate incoming header information,

wherein the host interface logic is coupled to receive and associate the incoming header information with the incoming payload data so as to generate an incoming data frame for delivery to the host processor.

14. A device according to claim 13, wherein the transmit protocol processor and the receive protocol processor are contained together in a single integrated circuit chip, and comprising a bus on the chip coupled to both the transmit and receive protocol processors.

15. A device according to claim 14, and comprising a shared memory, which is accessible to both the transmit and receive protocol processors via the bus.

16. A network interface device, comprising:

receive logic, which is coupled to receive from a network in accordance with a predetermined network protocol an incoming data packet comprising incoming data that includes an incoming header and incoming payload data, and which is arranged to select a header portion of the incoming data packet;

an incoming header memory, coupled to receive from the receive logic a header portion of the incoming data, which includes at least the incoming header;

an incoming data memory, coupled to receive from the receive logic a data portion of the incoming data, which includes at least the incoming payload data;

a receive protocol processor, coupled to read and process the header portion of the incoming data in accordance with the predetermined network protocol so as to generate incoming header information; and

host interface logic, which is coupled to receive and associate the incoming header information with the incoming payload data so as to generate an incoming data frame for delivery to a host processor.

17. A device according to claim 16, wherein the protocol comprises a network layer protocol.

18. A device according to claim 17, wherein the network layer protocol comprises an Internet Protocol (IP).

19. A device according to claim 16, wherein the protocol comprises a transport layer protocol.

20. A device according to claim 19, wherein the transport layer protocol comprises a Transport Control Protocol (TCP).

21. A device according to claim 19, wherein the transport layer protocol comprises a User Datagram Protocol (UDP).

22. A device according to claim 16, wherein the data memory and the header memory comprise parallel first-in-first-out (FIFO) buffers, which are arranged to hold the data portion and the header portion, respectively, for a plurality of frames of incoming data.

23. A device according to claim 22, wherein the header memory comprises a first FIFO buffer, coupled to hold the header portion before it is processed by the protocol processor, and a second FIFO buffer, coupled to receive the header information from the protocol processor and to deliver it to the transmit logic.

24. A device according to claim 22, wherein the header information comprises an instruction to the host interface logic, indicating a length of the payload data to read from the data portion in the data memory for inclusion in the incoming data frame.

25. A device according to claim 16, wherein the data portion of the incoming data comprises substantially all of the incoming data, and wherein the header information comprises an instruction to the host interface logic, indicating a length of the payload data to read from the data portion in the data memory for inclusion in the incoming data frame.

26. A device according to claim 16, wherein the receive logic comprises a control register, which is programmable with a length parameter, responsive to which the receive logic determines how many bits to select for inclusion in the header portion.

27. A device according to claim 16, wherein the outgoing header memory comprises a fast memory, coupled to the receive protocol processor so as to be accessed thereby in a single clock cycle of the processor.

28. A device according to claim 16, wherein at least the incoming data and header memories and the receive logic are contained together with the receive protocol processor in a single integrated circuit chip, and wherein the receive protocol processor is coupled to the host interface logic so as to enable reprogramming of the receive protocol processor.

29. A device according to claim 16, wherein the host interface logic comprises a direct memory access (DMA) engine, and wherein the receive protocol processor is arranged to generate DMA descriptors along with the incoming header information, so that the DMA engine writes the incoming data frame to a memory of the host processor responsive to the descriptors.

30. A method for transmitting data over a packet network, comprising:

receiving from a host processor a frame of outgoing data that includes outgoing header information and outgoing payload data;

writing the outgoing header information to an outgoing header memory;

writing the outgoing payload data to an outgoing payload memory, separate from the header memory;

reading and processing the outgoing header information from the outgoing header memory so as to generate at least one outgoing packet header in accordance with a predetermined network protocol; and

associating the at least one outgoing packet header with the outgoing payload data from the outgoing data memory, so as to generate at least one outgoing data packet for transmission over the network in accordance with the protocol.

31. A method according to claim 30, wherein the protocol comprises a network layer protocol.

32. A method according to claim 31, wherein the network layer protocol comprises an Internet Protocol (IP).

33. A method according to claim 30, wherein the protocol comprises a transport layer protocol.

34. A method according to claim 33, wherein the transport layer protocol comprises a Transport Control Protocol (TCP).

35. A method according to claim 33, wherein the transport layer protocol comprises a User Datagram Protocol (UDP).

36. A method according to claim 30, wherein writing the outgoing header information and writing the outgoing payload data comprise writing the information and the data to parallel first-in-first-out (FIFO) buffers for a plurality of frames of outgoing data in succession.

37. A method according to claim 36, wherein processing the outgoing header information comprises writing the at least one outgoing packet header to a further FIFO buffer in preparation for associating it with the outgoing payload data.

38. A method according to claim 30, wherein processing the outgoing header information comprises generating a plurality of outgoing packet headers, and wherein associating the at least one outgoing packet header with the outgoing payload data comprises associating each of the plurality of outgoing packet headers with a corresponding portion of the outgoing payload data so as to generate a sequence of outgoing data packets.

39. A method according to claim 30, wherein processing the outgoing header information comprises determining a data length parameter, and wherein associating the at least one outgoing packet header with the outgoing payload data comprises reading the data from the outgoing payload memory for inclusion in the outgoing packet responsive to the data length parameter.

40. A method for processing data received over a packet network, comprising:

receiving from a network in accordance with a predetermined network protocol an incoming data packet

comprising incoming data that includes an incoming header and incoming payload data;

writing a header portion of the incoming data packet to an incoming header memory, the header portion including at least the incoming header;

writing a data portion of the incoming data to an incoming data memory, separate from the incoming header memory, the data portion including at least the incoming payload data;

reading and processing the header portion of the incoming data from the incoming header memory in accordance with the predetermined network protocol so as to generate incoming header information; and

associating the incoming header information with the incoming payload data from the incoming data memory so as to generate an incoming data frame for delivery to a host processor.

41. A method according to claim 40, wherein the protocol comprises a network layer protocol.

42. A method according to claim 41, wherein the network layer protocol comprises an Internet Protocol (IP).

43. A method according to claim 40, wherein the protocol comprises a transport layer protocol.

44. A method according to claim 43, wherein the transport layer protocol comprises a Transport Control Protocol (TCP).

45. A method according to claim 43, wherein the transport layer protocol comprises a User Datagram Protocol (UDP).

46. A method according to claim 40, wherein writing the header portion and writing the data portion comprise writing the header portion and the data portion to parallel first-in-first-out (FIFO) buffers for a plurality of frames of incoming data.

47. A method according to claim 46, wherein processing the header portion comprises writing the incoming header information to a further FIFO buffer, in preparation for associating it with the incoming payload data.

48. A method according to claim 46, wherein writing the incoming header information comprises writing an instruction indicating a length of the payload data to read from the data portion in the data memory for inclusion in the incoming data frame.

49. A method according to claim 40, wherein writing the data portion of the incoming data comprises writing substantially all of the incoming data to the incoming data memory, and wherein processing the header portion comprises writing an instruction indicating a length of the payload data to read from the data portion in the data memory for inclusion in the incoming data frame.

50. A method according to claim 40, wherein writing the header portion comprises programming a control register with a length parameter, and determining how many bits to select for inclusion in the header portion responsive to the length parameter.

51. A method according to claim 50, wherein programming the control register comprises determining the length parameter based on a maximum header length permitted by the network protocol.

52. A method according to claim 40, wherein processing the header portion comprises generating a direct memory access (DMA) descriptor, and comprising writing the incoming data frame to a memory of the host processor responsive to the descriptor.